

## **Rapport 5eme séance :** **31/01/2022**

Projet : Med Dispenser

### **Avant la séance**

J'ai effectué des recherches sur le fonctionnement des servomoteurs et de l'afficheur I2C LCD 16x2. J'ai donc trouvé qu'il était impossible de faire un tour complet avec un servomoteur et qu'il existe deux méthodes avec lesquelles on peut coder le servomoteur : `write()` dont l'argument est un angle ou `writeMicroseconds()` dont l'argument est la durée de l'impulsion de commande du servomoteur. J'ai voulu commencer à faire le montage mais il me manquait des fils.

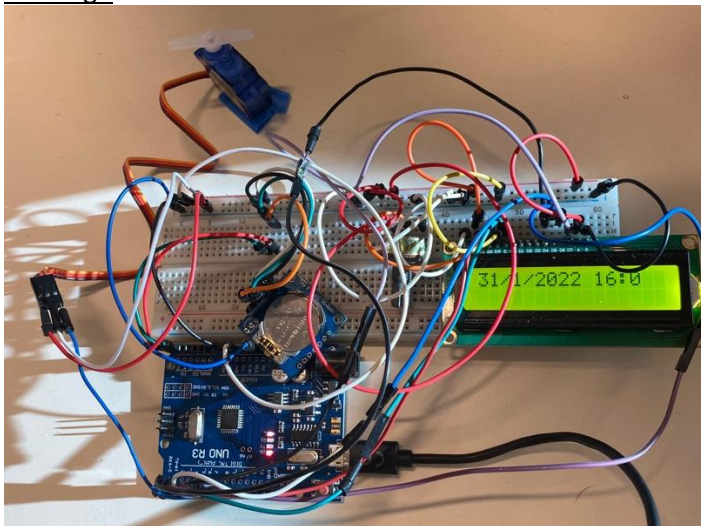
### **Durant la séance**

J'ai donc fait le montage d'un des servomoteurs ainsi que la rédaction du code. Celui-ci fonctionne, en effet le servomoteur tourne et fait des aller-retours.

J'ai ensuite modifié le code parent : j'ai supprimé le programme du moteur pas à pas pour rédiger celui du servomoteur, puis j'ai modifié le code pour actionner le moteur à l'heure souhaitée. Cela fonctionne.

J'ai ensuite voulu faire en sorte que le servomoteur fasse un seul aller-retour pour pouvoir plus tard choisir le nombre d'aller-retours qu'il fera mais je n'y suis pas parvenu.

### **Montage**



## Codes

```
codeparent  horlogeecran  servomoteur
#include <Servo.h>
Servo monServomoteur;
int position;

void initMotor() {
  monServomoteur.attach(9);
  position=0;
}

void runMotor() {
  // Fait bouger le bras de 0° à 180°
  for (int position = 0; position <= 180; position++) {
    monServomoteur.write(position);
    delay(15);
  }

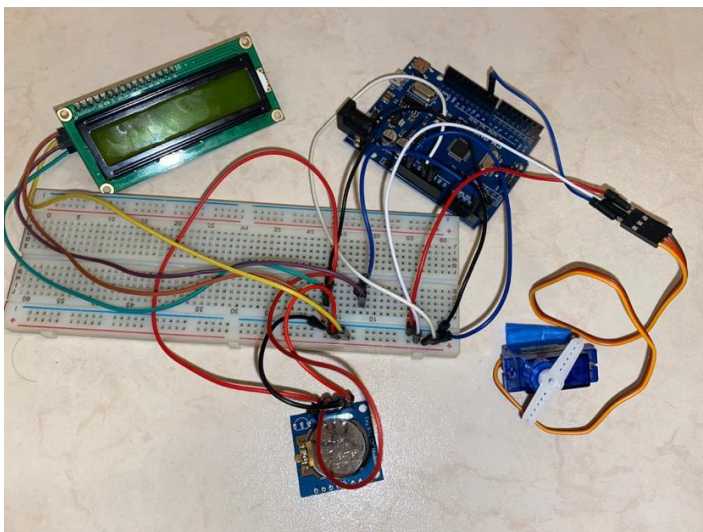
  // Fait bouger le bras de 180° à 10°
  for (int position = 180; position >= 0; position--) {
    monServomoteur.write(position);
    delay(15);
  }
  delay(2000);
}

codeparent  horlogeecran  servomoteur
//code principal
#include <RTClib.h>
RTC_DS1307 RTC;

void setup() {
  // put your setup code here, to run once:
  initHour();
  initMotor();
}

void loop() {
  // put your main code here, to run repeatedly:
  printHour();
  delay(3000);
  DateTime now=RTC.now();
  if ((now.hour()==17) && (now.minute()==47)){
    runMotor();
  }
  //runMotor();
}
```

Puis, j'ai fait le montage de l'afficheur I2C LCD 16x2, pour cela j'ai enlevé les anciens branchements (tous sauf celui du servomoteur). J'ai effectué des recherches pour trouver comment le brancher avec l'horloge mais selon les sites les montages n'étaient pas les mêmes et parfois l'horloge n'était pas la bonne. J'ai finalement réussi à faire ce montage grâce à certains sites et au cours.



J'ai ensuite refait le code, en effet avec cet afficheur il faut installer et utiliser une autre bibliothèque et le code n'est pas le même.

Or le nouveau code ne fonctionne pas : l'écran ne s'allume pas. J'ai donc cherché plusieurs exemples de codes sur internet pour comparer avec le mien et j'ai fait plusieurs modifications du code durant le temps de cours qu'il restait mais je ne suis pas parvenu à le faire marcher.

```
horloge_afficheur
#include <Wire.h>
#include <RTCLib.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C LCD(0x27,16,2);
RTC_DS1307 RTC;

void setup() {
  Serial.begin(9600);
  LCD.init();
  LCD.backlight();
  RTC.begin();
  LCD.begin(16,2);
  LCD.setCursor(0,1);
  /*Wire.begin();
  RTC.begin();
  LCD.begin(16,2);
  LCD.setCursor(0,1);*/
}

void loop() {
  DateTime now=RTC.now();
  LCD.clear();
  LCD.print(now.day(), DEC);
  LCD.print("/");
  LCD.print(now.month(), DEC);
  LCD.print("/");
  LCD.print(now.year(), DEC);
  LCD.print(" ");
  LCD.print(now.hour(), DEC);
  LCD.print(":");
  LCD.print(now.minute(), DEC);
}
```

Pour le prochain cours, je vais essayer de finir ce programme et ainsi d'allumer l'écran et je dois trouver une solution pour faire un seul aller-retour avec le servomoteur.